





Diamonds 0

Testing for Dungeon controller

```
Dungeon m = new Dungeon();
```

```
StringBuilder gameLog = new StringBuilder();
```

```
DungeonController c = new DungeonConsole(new StringReader(""), gameLog);
```

Construtor test	Inputs	output
Invalid input for row	-4	IllegalArgumentException
Invalid input for column	-4	IllegalArgumentException
Invalid input for treasure	-5	IllegalArgumentException
Invalid input for interconnectivity		IllegalArgumentException
testingMoveInput	Next possible move(s) are: NORTH, SOUTH, WEST Move, Pickup, or Shoot (M-P-S)? Move WEST	Player id: 1 is currently in location: 0, 4 You find 1 ruby here Treasures possessed are: 1. DIAMOND = 0 2. RUBIES = 0 3. SAPPHIRES = 1 4. ARROWS = 1 You find 1 arrow here Next possible move(s) are: NORTH, SOUTH, WEST Move, Pickup, or Shoot (M-P-S)?
testinginvalidMove	Next possible move(s) are: NORTH, SOUTH, WEST Move, Pickup, or Shoot (M-P-S)? M East	Please enter a valid move.
testingPickup	Next possible move(s) are: NORTH, SOUTH, WEST Move, Pickup, or Shoot (M-P-S)?	Treasures possessed are: 1. DIAMOND = 0 2. RUBIES = 0

	P	<p>3. SAPPHIRES = 1 4. ARROWS = 1 You find 1 arrow here</p> <p>Next possible move(s) are: NORTH, SOUTH, WEST</p> <p>Move, Pickup, or Shoot (M-P-S)?</p> <p>1. DIAMOND = 0 2. RUBIES = 0 3. SAPPHIRES = 1 4. ARROWS = 2 You find 1 arrow here</p>
testPlayerWinning		<p>Enter next move: N/n for NORTH, S/s for South, E/e for EAST and W/w for West</p> <p>n</p> <p>Game over!!</p>
testplayerEating		<p>Chomp, chomp, chomp, you are eaten by an Otyugh!</p> <p>Better luck next time</p>
testPlayerKilling		<p>You shoot an arrow into the darkness</p>
testingOutOfArrows		<p>Player id: 1 is currently in location: 0, 4 You find 1 ruby here</p> <p>Treasures possessed are:</p> <p>1. DIAMOND = 0 2. RUBIES = 0 3. SAPPHIRES = 1 4. ARROWS = 0</p> <p>You are out of arrows, explore to find more</p>

testingArrows		<p>Player id: 1 is currently in location: 0, 4 You find 1 ruby here</p> <p>Treasures possessed are:</p> <ol style="list-style-type: none"> 1. DIAMOND = 0 2. RUBIES = 0 3. SAPPHIRES = 1 <p>You find 1 arrow here</p> <p>Next possible move(s) are: NORTH, SOUTH, WEST</p> <p>Move, Pickup, or Shoot (M-P-S)?</p>
testingPungentSmell Test	<p>Next possible move(s) are: NORTH, SOUTH, WEST Move, Pickup, or Shoot (M-P-S)?</p> <p>M</p> <p>Where to? E</p>	<p>You smell something terrible nearby You are in a cave You find 1 arrow here</p>
testingShoot	<p>Move, Pickup, or Shoot (M-P-S)?</p> <p>S</p> <p>No. of caves (1-5)? 1</p>	<p>You shoot an arrow into the darkness</p>
treasurePickUp	<p>Next possible move(s) are: NORTH, SOUTH, WEST Move, Pickup, or Shoot (M-P-S)?</p> <p>P</p> <p>What? arrow</p> <p>You pick up an arrow</p>	<p>Treasures possessed are:</p> <ol style="list-style-type: none"> 1. DIAMOND = 0 2. RUBIES = 0 3. SAPPHIRES = 1 4. ARROWS = 1

hittingMonster	Move, Pickup, or Shoot (M-P-S)? S No. of caves (1-5)? 1	You hear a great howl in the distance
----------------	---	---------------------------------------

Testing suite for player class

```
Public void SetUp() {
    playerTest= new Player( 2);
```

Testing	Constraints	Input value	Expected value
constructor test		playertest	A player object with id 2
Invalid constructor test		playertest(-2)	IllegalArgumentExce ption
move(Direction)		Direction	Player is moved in any direction among the four enum types
getCurrentLocation			Returns the current location variable
getPossibleMoves			List of directions for the current cave.
+getTreasure()			List of Treasures
addTreasure(Treasure)		treasure	New treasure is added to the list of treasures.

Testing for cave class

```
Location caveTest = new cave();
```

Testing	Constraints	Input value	Expected value
constructor test		caveTest	An object of cave

			class
getTreasures			List of treasures
addTreasure(Treasure)		caveTest	A treasure from one the enum added.
removeTreasure		treasure	The given treasure is removed.

Testing for cave class

Location tunnelTest= new Tunnel();

Testing	Constraints	Input value	Expected value
constructor test		tunnelTest	An object of tunnel class
+getNeighbours()		tunnelTest	List of two neighbour locations

Testing Dungeon Class

MyRandom rand = new MyRandom(9,15,14,17,15,13,6,4,5,3,10,1,3,2,8,6,7,0,2,1,3,1,1,0,);

Dungeon myModel = new Dungeon(3, 4, 6, true, 20, rand);

Testing	constraints	Input value	Expected value
constructor test	Rand = 9	myModel	An object of the dungeon .
Invalid constructor test	Rand = 9	Dungeon(-3, 4, 6, true, 20, rand)	Illegalargumentexception
Invalid constructor test	Rand = 9	Dungeon(3, -4, 6, true, 20, rand);	Illegalargumentexception
Invalid constructor test	Rand = 9	Dungeon(3, 4, -6, true, 20, rand);	Illegalargumentexception

Invalid constructor test	Rand = 9	Dungeon(3, 4, 6, true, -20, rand);	Illegalargumentexception
addPlayer(Location)		Location	A player is added to the location.
getPotentialPaths : Set<Locations>		Location	A set of potential paths from a cave